The California Stream Condition Index (CSCI):
Interim instructions for calculating scores using GIS and R

Raphael Mazor[1,2] (raphaelm@scccwrp.org), Peter R. Ode[2],
Andrew C. Rehn[2], Mark Engeln[1], Tyler Boyle[3], Erik Fintel[3],
Steve Verbrugge[3], Calvin Yang[4] (calvin.yang@waterboards.ca.gov)

[1]Southern California Coastal Water Research Project. Costa Mesa, CA

[2]California Department of Fish and Wildlife. Rancho Cordova, CA

[3]Geographical Information Center, California State University. Chico, CA

[4]State Water Resources Control Board. Sacramento, CA

SWAMP-SOP-2015-0004

Revision Date: August 05, 2016

Table of Contents

Introduction

This document describes steps in calculating the California Stream Condition Index (CSCI), a bioassessment index that measures stream health based on benthic macroinvertebrate data. The instructions provided herein are provided as interim support for analysts requiring CSCI scores. The State Water Resources Control Board is currently developing a more automated approach to score calculation. Until that time, this document describes the only way to obtain CSCI scores.

The first section in this document describes the process for using a geographic information system (GIS) to calculate environmental predictors, such as watershed area and rainfall. The second section describes the process for using the environmental predictors, as well as taxonomic data, to calculate CSCI scores in R. A third section provides advice on interpreting scores in unusual circumstances (such as samples with poor taxonomic resolution).

The development and interpretation of the index is described in Mazor et al. (In press), which may be cited as follows:

> Mazor, R. D., P. R. Ode, A. C. Rehn, M. Engeln, K. A. Schiff, E. Stein, D. Gillett, D. Herbst, and C. P. Hawkins. In Press. Bioassessment in complex environments: Designing an index for consistent meaning in different settings. Freshwater Science.

A shorter summary of the index and its properties is available as a SWAMP technical memo:

> Rehn, A.C., R.D. Mazor and P.R. Ode. 2015. The California Stream Condition Index (CSCI): A New Statewide Biological Scoring Tool for Assessing the Health of Freshwater Streams. Swamp Technical Memorandum SWAMP-TM-2015-0002.

If you wish to cite this document to describe CSCI calculation (as opposed to general index properties or development), use the following citation:

> R. D. Mazor, P. R. Ode, A. C. Rehn, M. Engeln, T. Boyle, E. Fintel, S. Verbrugge, and C. Yang. 2015. The California Stream Condition Index (CSCI): Interim instructions for calculating scores using GIS and R. SCCWRP Technical Report #883. SWAMP-SOP-2015-0004.

Section 1: Instructions for Calculating CSCI Predictors with a Geographic Information System

The goal of this section is to guide users through the steps needed to calculate the predictors required for the California Stream Condition Index (CSCI).

These predictors are described as follows:

Predictor	Description
New_Lat	Latitude, in decimal degrees North
New_Long	Longitude, in decimal degrees East
SITE_ELEV	Site elevation in meters
ELEV_RANGE	Difference in elevation between the sample point and highest point in the catchment, in meters.
AREA_SQKM	Watershed area in square kilometers
PPT_00_09	Average precipitation (2000 to 2009) at the sample point, in hundredths of millimeters
TEMP_00_09	Average temperature(2000 to 2009) at the sample point, in hundredths of degrees Celsius
SumAve_P	Mean June to September 1971 to 2000 monthly precipitation, averaged across the entire catchment.
BDH_AVE	Average bulk soil density
KFCT_AVE	Average soil erodibility factor
P_MEAN	Average Phosphorous geology

Although the State Water Board will develop web-based tools to automate the steps described in this document, some users may be interested in calculating the CSCI on their own. We cannot guarantee the accuracy of metrics calculated using this document.

Field names and records are case-sensitive.

DOWNLOADING DATA

Temporarily, some of the necessary raster data may be downloaded in a compressed geodatabase on the SCCWRP FTP site:

ftp://ftp.sccwrp.org/pub/download/TMP/RaphaelMazor/CSCI_Predictor_Data.zip

This zip file contains a geodatabase, a python script for data consolidation and export, and documentation for each step in metric calculation. The documentation is redundant with the information in this SOP.

CREATING THE BASEFILES

BaseFiles are shapefiles that function as the unit of spatial analysis for calculation of CSCI predictors and other spatial metrics. The CSCI predictors are calculated with two types of BaseFiles: The site (a point representing the sample location) and a catchment (a polygon representing the contributing landuse). Reference screening requires several additional polygon BaseFiles, all of which are clipped from the catchment polygon: A 1-km clip, 5-km clip, and a 250-m riparian buffer.

All BaseFiles must contain a unique identifier of each station, which we call "StationCod" (this field name gets automatically changed to "StationCode" when data are exported for analysis in R). StationCods must be represented in all shapefiles, using the same letter case. Each StationCod must contain no more than 18 characters.

Creating the Site BaseFile

The goal of this step is to create a shapefile representing the location of sample points. Where possible, the location of sample points is adjusted ("snapped") from the actual coordinates to the nearest stream line represented in the National Hydrography Dataset Plus (NHD Plus). This "snapping" step is optional, and is recommended because it improves the catchment delineation process, and also to help generate metrics for screening reference sites. If snapping is not desired, stop after Step 4, but be sure to give subsequent delineations, metrics, and other analytical products additional scrutiny.

Data requirements

-Spreadsheet (e.g., in .xls format) with unique site identifiers (field name: StationCode) and coordinates in decimal degrees (fied names: LAT and LONG).

-NHD Plus layer (link), including flowlines and subbasins (Hydrologic Units)

1. Load spreadsheet in ArcMap.
2. Right-click and display XY data. X field is the LONG, Y field is the LAT. Set the Coordinate system to WGS84.
3. If you choose not to snap to an NHD Plus segment, add fields called "New_Lat" and "New_Long" that is identical to the original coordinates.
4. Reproject to NAD_1983_California_Teale_Albers.
5. Export the points and name them appropriately to make a permanent layer. Name this shapefile XXX_Sites (where XXX is the project name).

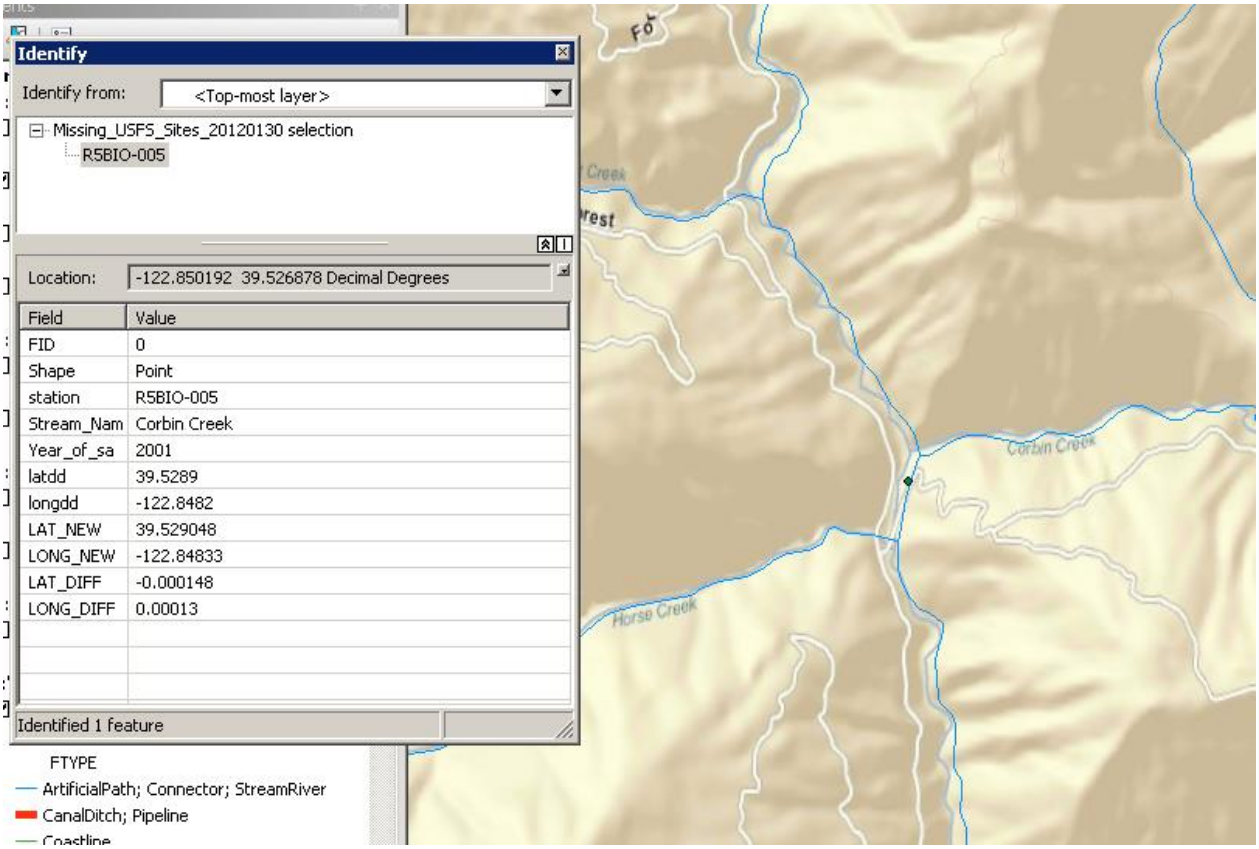
Snapping points to NHD flowlines:

6. Load the spatially corresponding NHD flowlines and the Subbasin from the Hydrologic Units folder. If all points fall within one NHD Region move on to the next steps. If not export them by region. They will need to be run separately through the delineator.
7. Snap the points to the nearest flowline in a manual edit session using "Edge Snapping". If there is no flowline near it, add a note in the attribute table. If the point is near a confluence or between two rivers look in the attribute table for clues to where it should go (see Quality Control section below).

8. Once all sites are snapped, add a "New_Lat" and "New_Long" field. Calculate the latitude and longitude of the newly snapped points.
9. Reproject to NAD_1983_California_Teale_Albers.
10. Export the points and name them appropriately to make a permanent layer. Name this shapefile XXX_Sites (where XXX is the project name).

Quality control for the Sites Basefile (for both snapped and unsnapped sites).

1. Ensure that snapped locations are reasonably close to reported sampling locations (generally, less than 0.003 decimal degrees, or ~300 m on the ground). Sites that snapped large distances should be flagged, so that the catchments delineated later can receive additional review. Large snapping distances are not always problems and may have minimal impact on the catchment or the metrics calculated from the BaseFiles. In a few cases it can actually lead to an improvement in the position of a site (e.g., if the original coordinates plotted on a mountain side and the shift moved them down into the channel).
2. Look for ancillary data (such as station names or descriptions, aerial imagery, USGS topographic maps) to verify sampling location. Contact sampling crews if necessary.
3. For sites close to confluences or near transitional areas, close scrutiny is required to ensure that the site is located on the correct stream segment. In the figure below, a site was sampled on Corbin Creek, near the confluence with the Eel River (as indicated by the site name). However, the point plots on the main stem of the Eel, downstream of the confluence. The coordinates needs to be manually corrected.



Creating the Catchments BaseFile

Below we outline the recommended approach for delineating catchments from a digital elevation model (DEM), simplified and improved by using pre-delineated watersheds in the National Hydrography Dataset Plus (NHD Plus). This approach works well for the majority of streams in California, although in certain situations alternative delineation methods may be preferable (particularly in flat areas with minimal topographic variation). No matter what approach is used, the goal is to identify the portion of the landscape that contributes runoff to a stream under natural ("reference") conditions. In general, dams, diversion, and inter-basin water transfers should be ignored when delineating the contributing catchment.

Requirements:

- Sites BaseFile
- 30-m DEM (link)
- NHD Plus (link)

1. Load sites BaseFile.
2. Load the NHD flowlines and Subbasins from the Hydrologic Units folder from the appropriate region; watersheds in different regions must be delineated in separate batches.
3. Save the sites attribute table as a tab-delimited text file.
4. Copy the text file to the NHDPlus Tools working directory on the processing computer
5. Start the basin delineator located here: http://www.horizon-systems.com/nhdplus/NHDPlusV2_tools.php#NHDPlusV2 BasinDelineator Tool
6. Click Run Basin Delineator.
7. The "Basin Pourpoints File" is the text file you made; browse to it.
8. Set the "Basin Shape Output File" to an appropriate directory.
9. Click Analyze, and watch for a pop-up when it completes. Depending on the number of catchments, delineation can take a long time.
10. After acknowledging the process has completed you may get a second pop-up saying that it was unable to delineate a number of catchments. You may have to delineate these manually.
11. Copy the output file back to your computer and load it into ArcMap, along with the local hydrology, catchments, HUCs, snapped points, and a base map.
12. Perform initial quality control checks (see section below).
13. Compare the delineated catchments to the sites BaseFile to find out which catchments have not been delineated.
14. Manually delineate those catchments that failed automatic delineation or were rejected during quality control checks in an edit session.
15. Recalculate the New_Lat and New_Long in case there were any changes to the point locations.
16. Once all catchments have been reviewed and delineated, set the data frame to NAD_1983_California_Teale_Albers.
17. Export the shapefile and name them appropriately to make a permanent layer. Name this shapefile XXX_WS (where XXX is the project name).

Quality Control Checks for Catchment Delineations

Helpful GIS files to support QC:

- NHD Plus stream network. You may want to hide pipelines, but keep canals visible with a distinct color. **Note: the NHD (1:24K) network is often needed to resolve discrepancies between NHD+ hydrology and DEM based hydrology. If there is a conflict, the 1:24K version is usually much more accurate.**
- Elevation files, shaded relief maps or topographic maps.

1. In general, it is best to examine each catchment individually. Highlighting (or selecting) each catchment, one at a time, makes many problems obvious.
2. Look for gross irregularities, such as:
    - Holes **Fix holes** (this is a pretty rare problem). Do this by removing the polygon vertices that create the hole.
    - Small nonsensical polygons that clearly don't correspond to a drainage network. These tend to occur when the coordinates plot off of a stream line and/or when the stream is in a flat area with little or no relief.
3. Does the watershed have a "lollipop" or "frying pan" shape? This problem is most common when the site is located in a flat area with few topographic features. Unless this shape is supported by the local topography, **flag the site for further review.** Use the catchments from the corresponding regions NHDPlus as guide to fixing "lollipops". Select and merge catchments to delineated catchments where necessary to fill out catchment, or manually correct.
4. For sites close to confluences (within ~300 m), make sure that the "correct" catchment was delineated. The only way to verify this may be to check the original site name or description, or to check with the original field crew that sampled the site.
5. Follow the perimeter of the delineation around the entire watershed. Note the following potential errors:
    - Does the delineation cross any ponds, reservoirs, or lakes? If so, does the topography support inclusion in/exclusion from the watershed? **Fix, or flag for further review**.
    - Do any NHD Plus flowlines cross the watershed border? If so, does the topography support inclusion in /exclusion from the watershed? Flowlines that represent pipelines, canals or aqueducts (or any situation where the flowline does not receive water from the immediate landscape) should be ignored. If necessary, check site with imagery from Google Earth. **Fix, or flag for further review**.
    - Most errors are small, and will have negligible influence on CSCI scores or other predictors. As a rule of thumb, errors can be ignored if they would modify the total area of the catchment <5%, and do not alter the type of landuse inside the delineation.
    - Watch for "divots" in the catchment. If the hydrology does not connect to the rest of the hydrologic network it will not be included in the catchment by the delineator even if they clearly feed into the catchment. Select the NHD Plus catchment and merge it into the delineated catchments in this case.

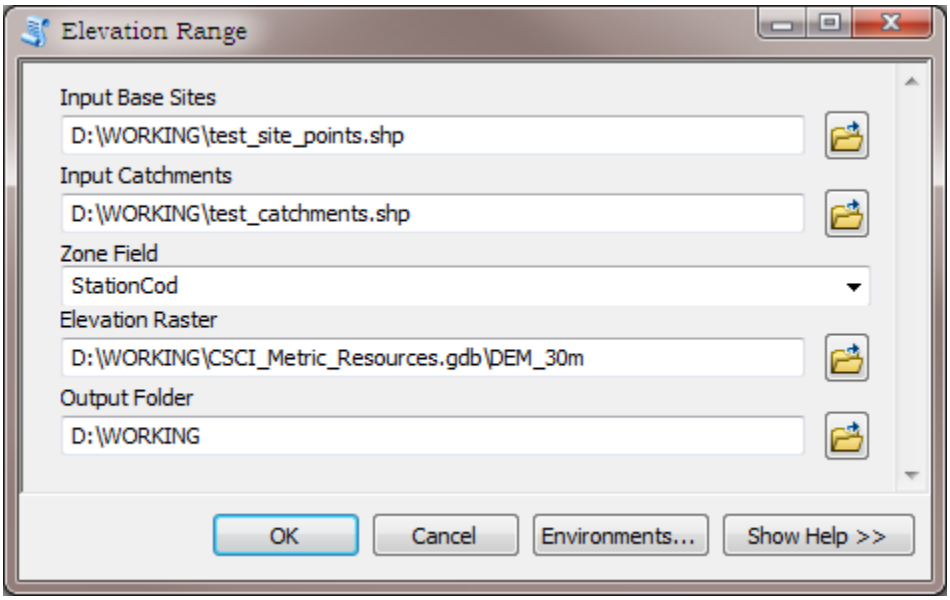
CALCULATING PREDICTOR METRICS

Elevation Metrics

The following describes how to process site elevation, watershed maximum elevation and the elevation change between them using the Elevation Range Python Tool in ArcGIS 10.0 and above. This tool requires the Spatial Analyst Extension to run.

Elevation Range Processing Tool

1. Navigate to the "CSCI_Metric_Toolbox" and double-click the "Elevation Range" script to open its dialog box.



Input Base Sites: Add the site points to this input. They should be in the format of the "Base_Sites_Template" feature class in the "CSCI_Metric_Resources" geodatabase (GDB).

Input Catchments: Add the catchments polygons to this input. The StationCode field must correspond with the Input Base Sites for the tool to run properly.

Zone Field: Choose the field that contains the unique id for each input catchment (e.g., StationCode). The input site points must also have this same field with the same set of unique id values.

Elevation Raster: This is the input DEM dataset. Add the "DEM_30m" raster located in the "CSCI_Metric_Resources" GDB.

Output Folder: Choose the location you wish the final results shapefiles to be saved. Intermediate files will also be saved here during processing but will be deleted upon completion.

2. Click "OK" and the tool will run. When it completes you should see two new shapefiles named "Catchments_Elevation_Ranges.shp" and "Sites_Elevation.shp".

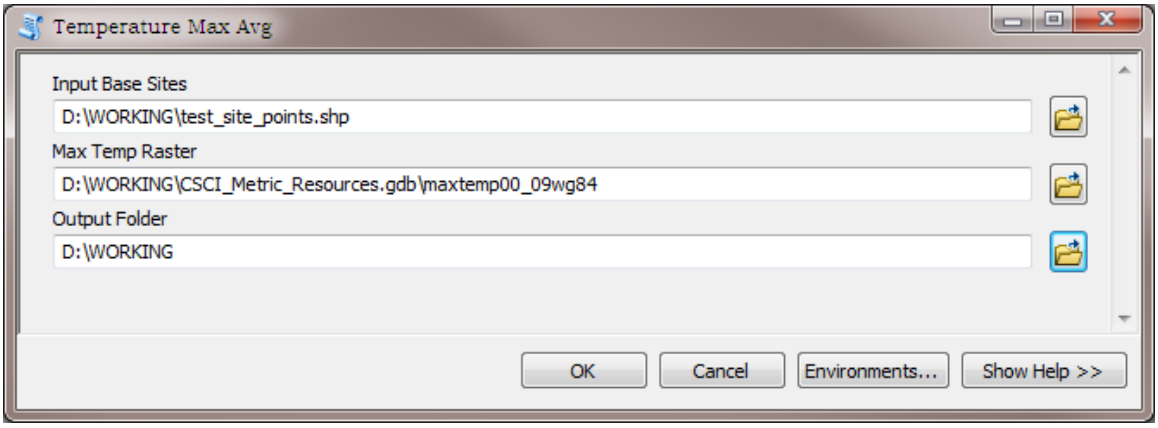
3. Add the new shapefiles to ArcMap and open the attribute tables.
    a. In the catchments attributes, the follow new fields are added.
        i. SITE_ELEV – Elevation at the sample site.
        ii. MAX_ELEV – Maximum elevation of the watershed.
        iii. ELEV_RANGE – Elevation range between sample site and top of watershed.
    b. In the site attributes, only the SITE_ELEV and ELEV_RANGE fields have been added.

Average Temperature

The following describes how to derive the average precipitation at a giving test site using the Temperature Avg Python Tool in ArcGIS 10.0 and above. This tool requires the Spatial Analyst Extension to run.

Temperature Processing Tool

1. Navigate to the "CSCI_Metric_Toolbox" and double-click the "Temperature Max Avg" script to open its dialog box.



Input Base Sites: Add the site points to this input. They should be in the format of the "Base_Sites_Template" feature class in the "CSCI_Metric_Resources" geodatabase (GDB).

Max Temp Raster: This is the input max temperature dataset. Add the "maxtemp00_09wgs84" raster located in the "CSCI_Metric_Resources" GDB.

Output Folder: Choose the location you wish the final results shapefile to be saved. Intermediate files will also be saved here during processing but will be deleted upon completion.

2. Click "OK" and the tool will run. When it completes you should see a new shapefile named "TempMaxAvg_00_09wgs84.shp". Add the new shapefile to ArcMap and open the attribute table. You will see that a new field "TEMP_00_09" has been added. It contains the maximum average temperature from 2000 to 2009 for each site.

   The temperature units are degrees Celsius multiplied by 100.

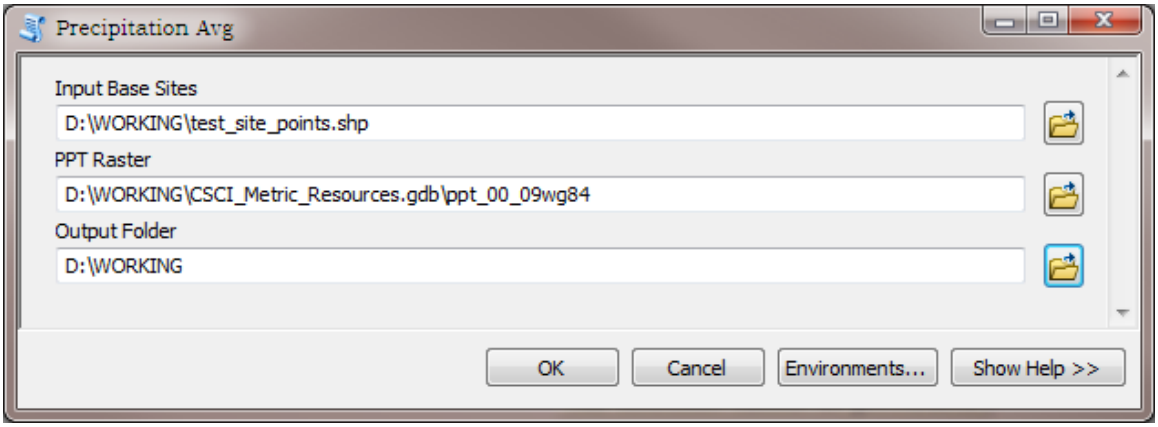
Notes: Currently to script output field assumes the standard 2000 to 2009 time frame but it can easily be modified to output a new field name for any time frame if new data is acquired.

Average Precipitation

The following describes how to derive the average precipitation at a giving test site using the Precipitation Avg Python Tool in ArcGIS 10.0 and above. This tool requires the Spatial Analyst Extension to run.

Precipitation Processing Tool

1. Navigate to the "CSCI_Metric_Toolbox" and double-click the "Precipitation Avg" script to open its dialog box.



Input Base Sites: Add the site points to this input. They should be in the format of the "Base_Sites_Template" feature class in the "CSCI_Metric_Resources" geodatabase (GDB).

PPT Raster: This is the input precipitation dataset. Add the "ppt_00_09wgs84" raster located in the "CSCI_Metric_Resources" GDB.

Output Folder: Choose the location you wish the final results shapefile to be saved.

2. Click "OK" and the tool will run. When it completes you should see a new shapefile named "PPTAvg_wgs84.shp". Add the new shapefile to ArcMap and open the attribute table. You will see that a new field "PPT_00_09" has been added. It contains the average precipitation from 2000 to 2009 for each site.

   The precipitation units are millimeters multiplied by 100.

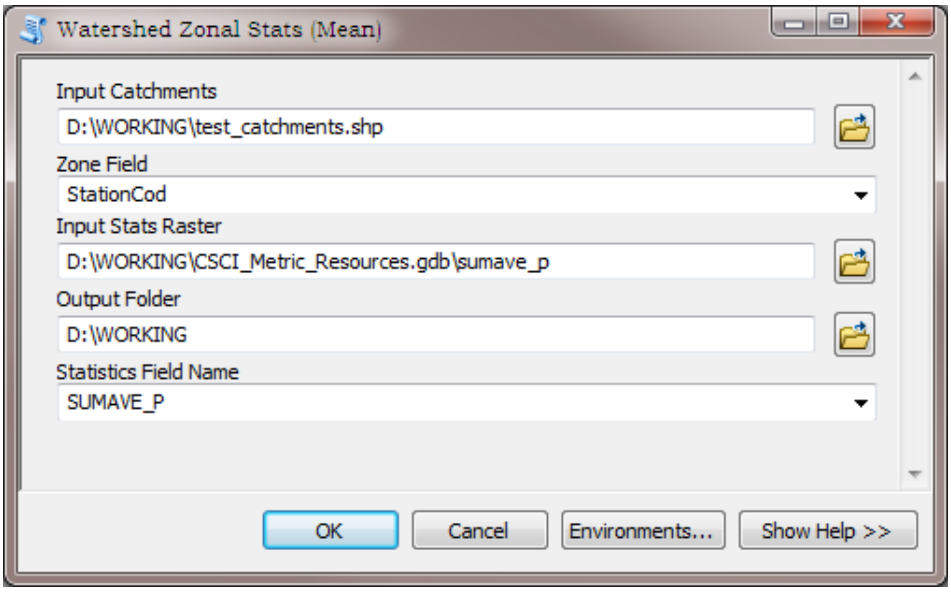
Notes: Currently to script output field assumes the standard 2000 to 2009 time frame but it can easily be modified to output a new field name for any time frame if new data is acquired.

Zonal Statistics (e.g., geology metrics)

The following section describes how to process average values of any input raster within a watershed using the Watershed Zonal Stats (Mean) Python Tool in ArcGIS 10.0. Currently, the tool is set up to work with only the predictors required for the CSCI (i.e., BDH_AVE, P_MEAN, SumAve_P, and KFCT_AVE), but may be expanded to other metrics in the future. The following tool requires that your data be projected in California NAD83 Teale Albers. This tool requires the Spatial Analyst extension to run.

Watershed Zonal Statistics Processing Tool

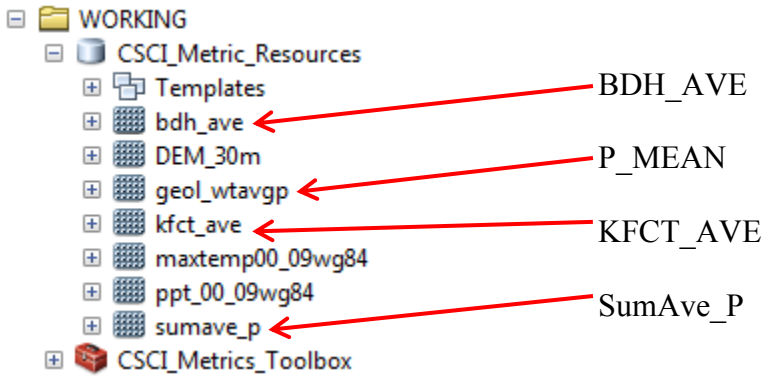
1. Navigate to the "CSCI_Metric_Toolbox" and double-click the "Watershed Zonal Stats (Mean)" script to open its dialog box.



Input Catchments: Add the catchments polygons to this input. They must be projected in California NAD83 Teale Albers

Zone Field: Choose the field that contains the unique id for each input catchment (e.g., "StationCode"). *NOTE: The unique ID value for each catchment cannot exceed 18 characters in length or the tool will fail. Check your data beforehand and shorten the IDs if necessary.

Input Stats Raster: Navigate to the "CSCI_Metric_Resources" geodatabase (GDB) and select the raster that corresponds with the metric you would like to calculate. See figure below.

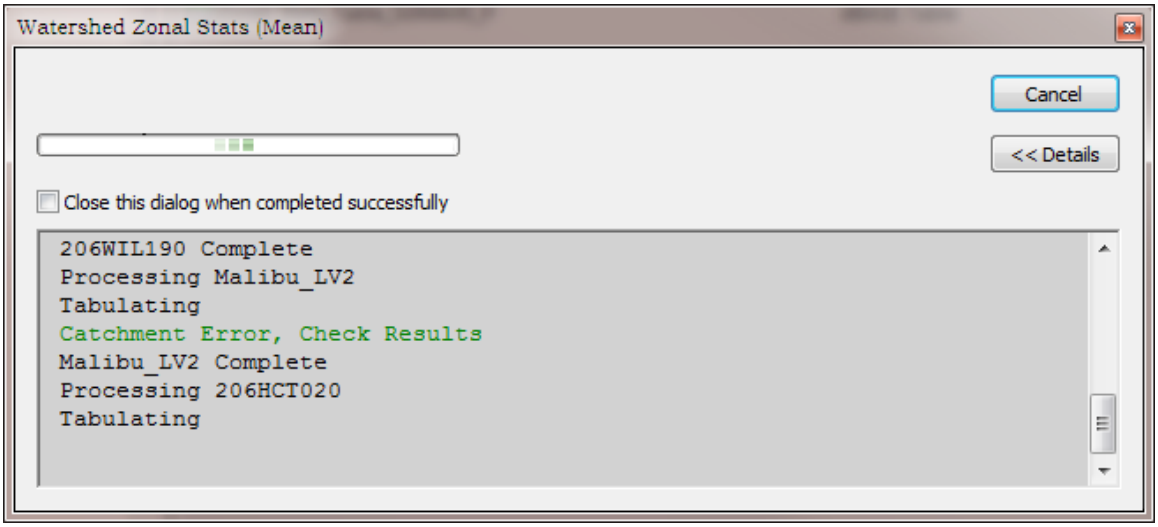


In this example we will calculate SumAve_P.

Output Folder: Choose the location you wish the final results shapefiles to be saved. Intermediate files will also be saved here during processing but will be deleted upon completion.

Statistics Field Name: From the drop down menu, choose the metric you wish to calculate. This will become the name of the output field in the resultant output dataset.

2. Click "OK" and the tool will run. As the tool runs, the name of the catchment being processed is displayed in the dialog. In some cases a catchment will not overlap with the input raster, or is too small compared with the input raster cell size. In these cases, the message "Catchment Error, Check Results" is displayed.

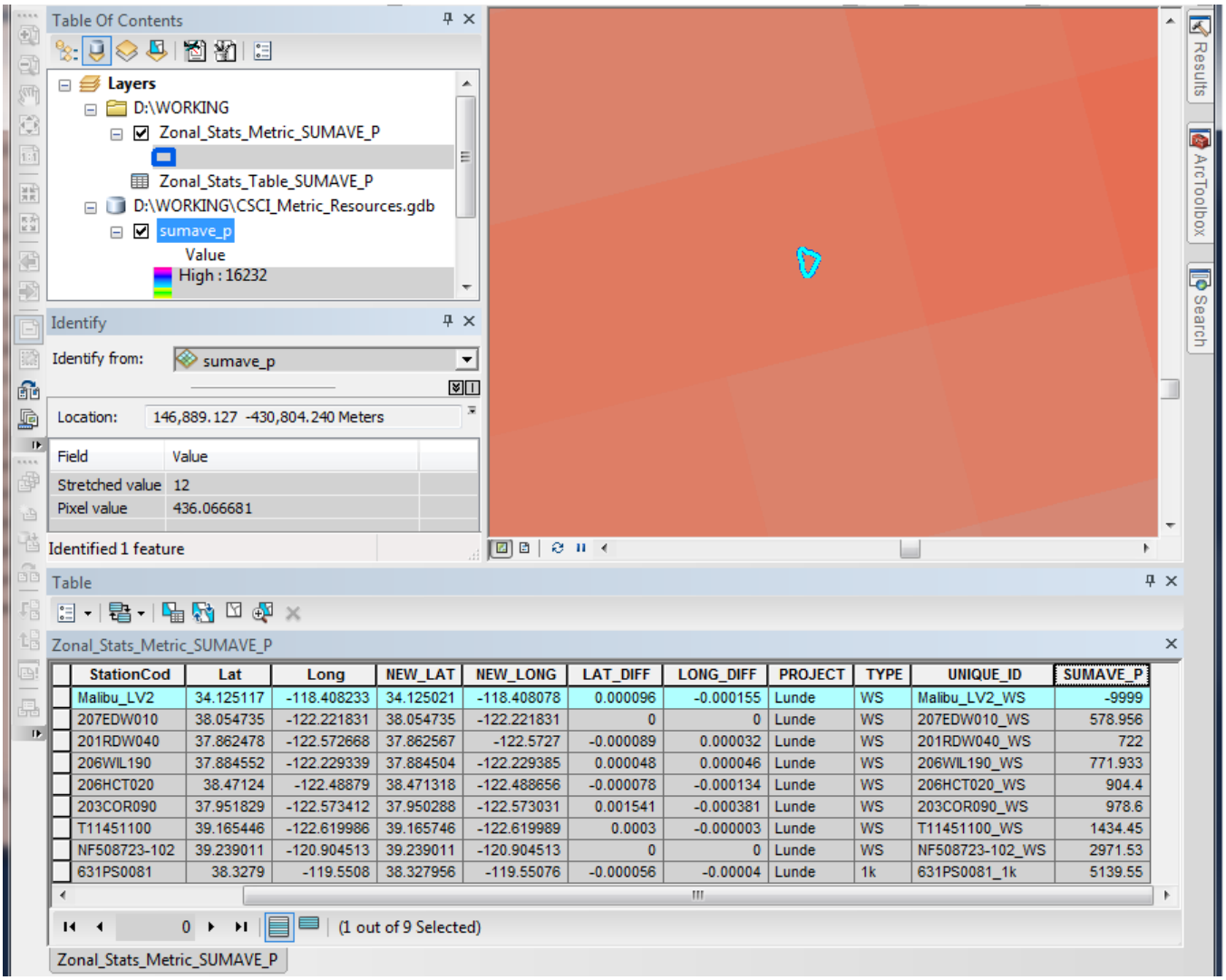


In this example, the catchment "Malibu_LV2" did not process in the model properly. It will be assigned the value -9999. Each catchment given the value -9999 will need to be manually reviewed to determine proper action. This process will be explained in more detail in steps 4 through 7.

3. When the tool completes, two output files have been created. They are "Zonal_Stats_Metrics_SUMAVE_P.shp" and "Zonal_Stats_Table_SUMAVE_P.dbf" in this example. The Statistics Field Name chosen is used to name your output file. Add both files to ArcMap and open their attribute tables. Both the shapefile and table have output statistics metric, in this case "SUMAVE_P". The table output is strictly for convenience if a shapefile is not desired.

4. Now go to the "Zonal_Stats_Metrics_SUMAVE_P.shp" attributes and sort ascending on the "SumAve_P" field. Check for any values of -9999. If none are present then your data is complete. In this example catchment "Malibu_LV2" was assigned the value -9999.

5. Add the Input Stats Raster to ArcMap and zoom to "Malibu_LV2". In the example below we can see that the catchment was too small for the Zonal Statistics operation to run properly against the "sumave_p" raster.



StationCod	Lat	Long	NEW_LAT	NEW_LONG	LAT_DIFF	LONG_DIFF	PROJECT	TYPE	UNIQUE_ID	SUMAVE_P
Malibu_LV2	34.125117	-118.408233	34.125021	-118.408078	0.000096	-0.000155	Lunde	WS	Malibu_LV2_WS	-9999
207EDW010	38.054735	-122.221831	38.054735	-122.221831	0	0	Lunde	WS	207EDW010_WS	578.956
201RDW040	37.862478	-122.572668	37.862567	-122.5727	-0.000089	0.000032	Lunde	WS	201RDW040_WS	722
206WIL190	37.884552	-122.229339	37.884504	-122.229385	0.000048	0.000046	Lunde	WS	206WIL190_WS	771.933
206HCT020	38.47124	-122.48879	38.471318	-122.488656	-0.000078	-0.000134	Lunde	WS	206HCT020_WS	904.4
203COR090	37.951829	-122.573412	37.950288	-122.573031	0.001541	-0.000381	Lunde	WS	203COR090_WS	978.6
T11451100	39.165446	-122.619986	39.165746	-122.619989	0.0003	-0.000003	Lunde	WS	T11451100_WS	1434.45
NF508723-102	39.239011	-120.904513	39.239011	-120.904513	0	0	Lunde	WS	NF508723-102_WS	2971.53
631PS0081	38.3279	-119.5508	38.327956	-119.55076	-0.000056	-0.00004	Lunde	1k	631PS0081_1k	5139.55

6. Use the Identify tool to determine the value of the raster at "Malibu_LV2"; in this case it's 436.066681. Use the field calculator to replace -9999 with the raster cell value.

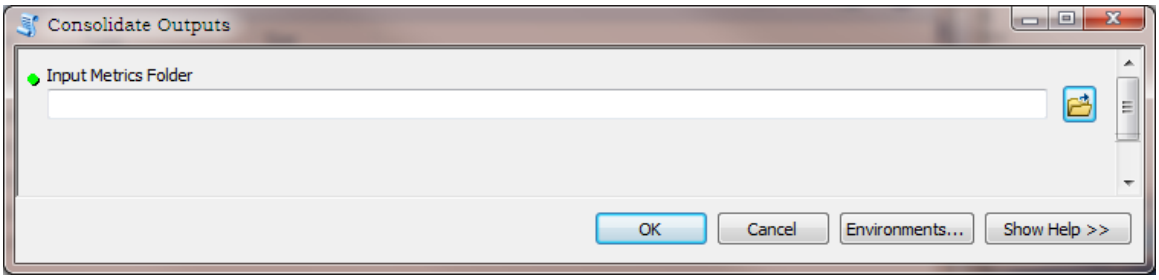
7. Repeat this process for any catchment that has a value of -9999. If the catchment does not overlap with the raster, assign the value from the raster cell closest to the catchment.

METRIC CONSOLIDATION AND DATA EXPORT

The following describes how to merge all results from other tools into a single CSV file ready for input into the R model using the Consolidate Outputs Python Tool in ArcGIS 10.0 and above.

Consolidate Metrics Processing Tool

1. Navigate to the "CSCI_Metric_Toolbox" and double-click the "Consolidate Outputs" script to open its dialog box.



2. Click "OK" and the tool will run. When the tool completes a CSV file named "Final_Metrics_Consolidated.csv"

If any CSCI Metrics are missing from the chosen folder location, and error message will be given indicating which metric files are missing. Add the required files to the folder and rerun the tool.

Section 2: Instructions for Calculating CSCI Scores in R

This document assumes that the user is familiar with basic operations in the R programming language, such as data import, export, and manipulation. Although not required, we recommend using graphic interface for R, such as R-studio, which can be downloaded at http://www.rstudio.com. New users are encouraged to pursue training opportunities, such as those hosted by local R user groups. A list of such groups may be found here: http://blog.revolutionanalytics.com/local-r-groups.html.

This document describes usage of CSCI package version 1.1.2.

THE SHORT VERSION

```
#Install the CSCI package the first time you run this
install.packages("devtools") #Install devtools from CRAN
library(devtools)
install_github("SCCWRP/BMIMetrics")
install_github("SCCWRP/CSCI")

#Load the library
library(CSCI)

#Import the bugs and stations data
bugs.df<-read.csv("bugs.csv")
stations.df<-readcsv("stations.csv")

#Optional: Clean the bugs data if life stage codes are bad or missing
bugs.df<-cleanData(bugs.df)

#Calculate the CSCI
#Optional rand argument makes results repeatable
report<-CSCI(bugs.df, stations.df, rand=1)

#Export the desired reports
write.csv(report$core, "core.csv")#CSCI component scores, basic data
quality info
write.csv(report$Suppl1_mmi, "Suppl1_mmi.csv")#Details about pMMI
score
write.csv(report$Suppl1_grps, "Suppl1_grps.csv") #Details on ref group
membership
write.csv(report$Suppl1_OE, "Suppl1_OE.csv") #Details about O/E score
write.csv(report$Suppl2_mmi, "Suppl2_mmi.csv") #Iteration-level
details on pMMI score
write.csv(report$Suppl2_OE, "Suppl2_OE.csv")#Iteration-level details
on O/E score
```

THE DETAILED GUIDE

Installing R-scripts

Make sure you have a good internet connection, and then run this line in the R console:

```
install.packages("devtools") #Install devtools from CRAN
library(devtools)
install_github("SCCWRP/BMIMetrics")
install_github("SCCWRP/CSCI")
```

These lines will automatically install the `CSCI` package, as well as its dependent packages (`randomForest`, `vegan`, `stringr`, `reshape2`, `plyr`, and `data.table`). This process may take several minutes because the models and data tables required for the CSCI are fairly large (~100 MB). You may get a warning about the file size mismatching its reported length, but this warning may be disregarded.

If installation is successful, you should be able to launch the CSCI library and access the help pages:

```
library(CSCI)
?CSCI
```

To receive alerts about package updates, you may join the CSCI users listserve by emailing Raphael Mazor (raphaelm@sccwrp.org). This listserve will eventually be replaced by one maintained by SWAMP.

Preparing the Input Data

Stations Data

Stations data includes all the environmental information for each station, with one row per station. The required fields are:

Field Name	Description
StationCode	Unique identifier of the site
New_Lat	Latitude in decimal degrees
New_Long	Longitude in decimal degrees
SITE_ELEV	Site elevation
ELEV_RANGE	Difference in elevation between the sample site and the highest point in the catchment
AREA_SQKM	Area of the catchment
TEMP_00_09	Long-term mean temperature at the site
PPT_00_09	Long-term mean precipitation at the site
SumAve_P	Mean summer precipitation across the catchment
KFCT_AVE	Average soil erodibility factor
BDH_AVE	Average soil bulk density
P_MEAN	Phosphorous content of the catchment geology

Field names must match spelling shown above. For the required fields, blank cells or missing values are not allowed. Please see Section 1 for information on calculating predictor data. Other fields of interest may be included in the stations data. Columns may appear in any order. Although we have implemented scripts to make the inputs case-insensitive, we recommend conforming to the capitalizations shown above.

An example of properly formatted stations data is included in the package:

```
data(bugs_stations)
stations<-bugs_stations[[2]]
```

Bugs Data

Bugs data includes all the taxonomic information for each sample, with one row per taxon (that is, flat-file format). The required fields are:

Field Name	Description
StationCode	Unique identifier of the site
SampleID	Unique identifier of the sample. Recommended format: A concatenation of StationCode, sample date, collection method code, and field replicate number.
FinalID	Taxonomic names. Must match values in SWAMP organism lookup lists (http://swamp.waterboards.ca.gov/SWAMP_Checker/DisplayLookUp.php?List=OrganismDetailBMILookUp). The match is not case sensitive, and a few common misspellings are recognized.
Distinct	~~Indicator of distinct taxa, provided by taxonomist. Use positive integers to indicate distinct taxa. Optional. OK to leave blank (or NA) for unknowns.~~ UPDATE: We recommend that in all cases, this field be left blank for every row of the input data.
LifeStageCode	Indicator of life stages: A for adult insects, L for larval insects, P for pupal insects, and X for non-insects. Not case-sensitive. All combinations of FinalID and LifeStageCode must be found in SWAMP organism detail lookup lists: http://swamp.waterboards.ca.gov/SWAMP_Checker/DisplayLookUp.php?List=OrganismDetailBMILookUp. If unknown or uncertain, you can use the cleanData() function, described below.
BAResult	Total count of the organisms

Field names must match spelling shown above. Except for Distinct and LifeStageCode, blank cells or missing values are not allowed. All StationCodes used in the bugs file must also appear in the stations file, and vice-versa. Columns may appear in any order. Although we have implemented scripts to make the inputs case-insensitive, we recommend conforming to the capitalizations shown above.

Getting taxonomy data from SWAMP:
If you have access to the SWAMP Reporting Module, query your benthic data as you normally would. Go to "BMI Base Queries" and export the "Benthic Taxonomy Results" report as a csv. This report should be properly formatted for calculating the CSCI.

Getting taxonomy data from CEDEN:

Benthic macroinvertebrate data are available to the general public through CEDEN (www.ceden.org). Although queries on CEDEN allow the downloading of benthic macroinvertebrate data, users will need to manually select results related to stream benthic macroinvertebrate samples (as opposed to data related to fish, algae, or plants, or non-stream macroinvertebrates). Additionally, life stage and distinct information is provided by CEDEN, but these data will require reformatting to meet the requirements of the CSCI package. The `cleanData` function and the `purge` argument (described below) may be helpful in refomatting data downloaded from CEDEN.

Calculating the CSCI

Overview:

The `CSCI` package automates all of the necessary steps to calculate CSCI scores from properly formatted input files. It uses takes the predictor data in the stations input file to calculate biological expectations using random forest models. It uses the biological data in the bugs input file to calculate metrics and other biological endpoints. Additionally, it compares the endpoints to the expectations, relative to a reference distribution. We have automated many of these steps, with the goal of minimizing demands on the user.

The automated steps are as follows:

For O/E calculation:

1. Aggregate taxa to operational taxonomic units (OTUs).
2. Exclude ambiguous taxa (e.g., taxa identified to relatively poor taxonomic resolution).
3. For samples with more than 400 remaining specimens, subsample to 400 specimens (20 iterations).
4. Use stations data to predict group membership and calculate OTU capture probabilities.
5. Calculate O/E score for each iteration, using a minimum capture probability of 0.5.

For pMMI calculation:

1. Aggregate taxa to SAFIT Level 1.
2. For samples with more than 500 remaining specimens, subsample to 500 specimens (20 iterations).
3. Calculate biological metrics.
4. Use stations data to predict metric values.
5. Calculate difference between observed and predicted metric values. Score the difference, calculate the average across metrics, and standardize by dividing by the mean from reference calibration sites (i.e., 0.628).

For CSCI calculation:

1. Calculate the average O/E and pMMI scores, as described above.
2. Compare the CSCI, O/E, and pMMI scores to the distribution of scores at reference calibration sites.

Note that there are two distinct subsampling steps (i.e., for the O/E and for the pMMI), and each are triggered by different criteria. The number of iterations for each subsampling step is provided in the reports.

Caveats:

Many steps typically required of index calculation are hardwired into the scripts, and are automatically handled. Specifically, FinalIDs are aggregated to the necessary taxonomic resolution, and large samples are subsampled to the required size. We strongly discourage all efforts to manually aggregate or subsample your own data, and instead recommend you rely on the standardized, automated approach implemented by the provided scripts.

Getting your score:

To calculate the CSCI, first load your bugs and stations data into the workspace, and load the CSCI library:

```
bugs.df<-read.csv('bugs.csv')
stations.df<-read.csv('stations.csv')
library(CSCI)
```

The CSCI function will calculate scores from the bugs and stations data:

```
report<-CSCI(bugs=bugs.df, stations=stations.df)
```

There are only two required arguments for the `CSCI()` function: bugs and stations. Optional arguments include the following:

`rand:` Specify an integer to set the random seed, thereby ensuring that the subsampling procedure can be replicated on repeated runs of the script. By default, set to `sample.int(1000, 1)`.

`purge:` Automatically excludes all FinalID/LifeStageCode combinations that do not match associated lookup lists. If TRUE, purged taxa will be listed in the output. If FALSE (default), any unrecognized combinations will cause an error. We recommend resolving mismatches of FinalID/LifeStageCode by reviewing the data, and not by using the purge argument; however, we provide it as a shortcut for data analysis.

Interpreting the outputs:

The `CSCI()` function produces 6 reports, each as a named dataframe within a list. They can be accessed using normal R indexing (e.g., report$core, report$Suppl1_mmi, etc.). The reports are summarized as follows:

Report Component	Description
core	A summary of the CSCI results and data quality flags, averaged across 20 iterations.
Suppl1_mmi	A detailed breakdown of the pMMI component of the CSCI. Raw, predicted, and scored metric values, averaged across 20 iterations.
Suppl1_grps	Probability of biotic group membership, with one row per SampleID.
Suppl1_OE	A detailed breakdown of the O/E component of the CSCI. OTU capture probabilities and mean abundances, averaged across 20 iterations.
Suppl2_mmi	Similar to Suppl1_mmi, except with results for each iteration provided.
Suppl2_OE	Similar to Suppl1_OE, except broken down by iteration. Iteration-wise O/E scores are also provided.

Field definitions for each report are provided below:

Core report

Field Name	Description
StationCode	Unique identifier of the site
SampleID	Unique identifier of the sample
Count	Total number of organisms in the sample. If purge=T, the post-purge number is shown. A minimum number has not been established, but samples with low values should be evaluated with caution.
Number_of_MMI_Iterations	Number of subsamples used to calculate the pMMI. If the count is less than 500, no subsampling is performed, and this field will show 1. Otherwise, 20 subsamples are performed.
Number_of_OE_Iterations	Number of subsamples used to calculate the O/E. If the total number of unambiguous taxa is less than 500, no subsampling is performed, and this field will show 1. Otherwise, 20 subsamples are performed.
Pcnt_Ambiguous_Individuals	Percent of the total number of individuals excluded from O/E calculation. A maximum number has not been established, but samples with high values should be evaluated with caution.
Pcnt_Ambiguous_Taxa	Percent of the total number of FinalIDs excluded from O/E calculation. A maximum number has not been established, but samples with high values should be evaluated with caution.
E	The sum of all capture probabilities greater than 0.5 at a site. Interpreted as the total number of common taxa expected at a site.
Mean_O	The number of common taxa (i.e., capture probability greater than 0.5) observed at a site, averaged across iterations.
OoverE	O/E, calculated as Mean_O divided by E.
OoverE_Percentile	The percentile of the O/E score, relative to the reference distribution. A minimum threshold has not been established, but low values should be considered indicative of degradation.
MMI	The pMMI score, averaged across 20 iterations. A minimum threshold has not been established, but low values should be considered indicative of degradation.
MMI_Percentile.	The percentile of the pMMI score, relative to the reference distribution. A minimum threshold has not been established, but low values should be considered indicative of degradation.
CSCI	The CSCI score, calculated as the average of the O/E and pMMI.
CSCI_Percentile	The percentile of CSCI score, relative to the reference distribution. A minimum threshold has not been established, but low values should be considered indicative of degradation.

Suppl1_mmi. All values are averaged across 20 iterations.

Field Name	Description
StationCode	Unique identifier of the site
SampleID	Unique identifier of the sample
MMI_Score	pMMI score
Clinger_PercentTaxa	Observed percent clinger taxa
Clinger_PercentTaxa_predicted	Predicted percent clinger taxa
Clinger_PercentTaxa_score	Score for percent clinger taxa metric
Coleoptera_PercentTaxa	Observed percent Coleoptera taxa
Coleoptera_PercentTaxa_predicted	Predicted percent Coleoptera taxa
Coleoptera_PercentTaxa_score	Score for percent Coleoptera taxa metric
Taxonomic_Richness	Observed taxonomic richness
Taxonomic_Richness _predicted	Predicted taxonomic richness
Taxonomic_Richness_score	Score for taxonomic richness metric
EPT_PercentTaxa	Observed percent Ephemeroptera, Plecoptera, and Trichoptera (EPT) taxa
EPT_PercentTaxa_predicted	Predicted percent EPT taxa
EPT_PercentTaxa_score	Score for EPT percent taxa metric
Shredder_Taxa	Observed number of shredder taxa
Shredder_Taxa_predicted	Predicted number of shredder taxa
Shredder_Taxa_score	Score for shredder taxa metric
Intolerant_percent	Observed percent intolerant individuals (CTV<3)
Intolerant_percent_predicted	Predicted percent intolerant individuals
Intolerant_percent_score	Score for percent intolerant individuals metric

Suppl1_grps

Field Name	Description
StationCode	Unique identifier of the site
pGroupX	Probability that site is a member of group X.

Suppl1_OE

Field Name	Description
StationCode	Unique identifier of the site
SampleID	Unique identifier of the sample
OTU	Operational taxonomic unit. All OTUs with capture probability greater than 0 are shown, but only those with a capture probability greater than 0.5 are used for scoring.
CaptureProb	Probability of observing the OTU at the site.
Mean Observed	Number of individuals observed in the sample, averaged across 20 iterations

Suppl2_mmi

Field Name	Description
StationCode	Unique identifier of the site
SampleID	Unique identifier of the sample
metric	Name of the metric
Iteration	Unique identifier of the iteration
value	Observed metric value for each iteration
predicted_value	Predicted metric value. Same for all iterations.
score	Scored difference between predicted and observed value for each iteration of metric

Suppl2_OE

Field Name	Description
StationCode	Unique identifier of the site
SampleID	Unique identifier of the sample
OTU	Operational taxonomic unit. Unlike Supplement 1, all OTUs are shown. Also, the O/E score for each iteration is shown where the OTU is "OoverE."
CaptureProb	Probability of observing the OTU at the site.
IterationX	Number of individuals observed in Iteration X

Accessing Metadata and Reference Data

The CSCI package includes two built-in functions to give interested users access to some helpful information about the CSCI.

The `loadMetaData()` function generates a table containing all recognized species names (including a few common misspellings). This table is used to aggregate to SAFIT Level II or to OTUs, and to assign functional feeding groups, tolerance values, and other life history information used in metric calculation.

The `loadRefData()` function generates a table containing reference data used to calibrate the CSCI. Specifically, it includes the name of each reference site, sample dates, scores, biotic group membership, and predictor values.

TROUBLESHOOTING AND FAQ

Most problems result from errors in data formatting, or other errors in the input data. Most errors will prevent complete execution of the `CSCI()` function. We have attempted to provide informative error messages to help guide corrections.

Bad or missing field names

All required field names must be present in input files. Please be sure to match the field names provided above. Although we have implemented scripts to make the inputs case-insensitive, we recommend conforming to the capitalizations shown above.

Bad or missing life stage codes

If your data are missing life stage codes, or contain values that do not match acceptable values in SWAMP, we recommend the following assumptions:

- All non-insects are X
- All Hydraenidae and Hydrophilidae are A
- All other insects are L

To automatically implement these assumptions on records that do not have acceptable life stage codes, you can use the `cleanData()` function:

```
bugs2<-cleanData(bugs)
```

Missing data

With few exceptions, missing values are not allowed.

Bad FinalIDs

Bad FinalIDs typically result from misspellings, but occasionally occur when taxonomists do not conform to SAFIT's standard taxonomic effort (available at http://safit.org/ste.html). If your data set has incorrect bug names, you may use the `purge=T` argument in the `CSCI()` function. This allows calculation of CSCI scores even if the input data has unrecognized taxa. However, it is always preferable to correct the names than to purge them, and the purge argument should only be used for preliminary analyses.

If you believe a FinalID is erroneously missing from SWAMP's lookup lists, please contact the SWAMP help desk (OIMA-Helpdesk@waterboards.ca.gov). If you believe a valid FinalID is inappropriately rejected by the scripts, contact Raphael Mazor at raphaelm@sccwrp.org.

The `loadMetaData()` function provides a containing all recognized names, which may help identify misspellings or other problems creating errors. Please check this table before submitting a request for a modification to the script.

Importing characters as factors

R may import character vectors (like FinalID) as factors, which may not be interpreted correctly. We recommend importing all text fields as characters:

```
my.data.frame<-read.csv("myfile.csv", stringsAsFactors=F)
```

or coercing them into character format:

```
my.data.frame$FinalID<-as.character(mydata.frame$FinalID)
```

Stations that are very close together

If you are scoring two stations that are so close together that the GIS data look identical, the CSCI function may produce an error. There are two easy work-arounds you may use in this situation: 1) Remove one of the redundant rows from the stations data, and treat the two samples as though they were coming from the same stations. 2) Increase the precision of at least one GIS variable so they no longer appear identical (e.g., 5 or more decimal points).

Need more help?

Join the CSCI users listserve by emailing Raphael Mazor (raphaelm@sccwrp.org). This listserve will eventually be replaced by one maintained by SWAMP.

Stations that are in Mexico

Portions of some streams include areas in Mexico. Because the geodatabases used to calculate CSCI predictors do not currently include this area, the CSCI cannot be calculated properly for these sites. The geodatabases will be updated within the next few months. In the interim, we make the following recommendations: If more than 90% of the area of a watershed is within California, treat the state boundary as the edge of the watershed and calculate the predictors accordingly. However, you should interpret these results with caution, particularly if the portion within Mexico contains substantially different natural features. For watersheds that are less than 90% within California, we recommend using the Southern California Index of Biotic Integrity (Ode et al. 2005) as a substitute index. Additionally, indices based on benthic algae (see Fetscher et al. 2014) may also be calculated in these streams.

I want to calculate the SoCal IBI/NorCal IBI, etc. Can I do that with the CSCI package?

No, but the CSCI package can make the calculations easier. There's no automatic feature to allow you to calculate any of the old IBIs (although we may add that in a future version). However, there are some functions embedded within the CSCI package that you can use to calculate the metrics. Scoring and IBI calculation could subsequently be done by hand, as per IBI requirements.

```
library(CSCI)
#Import the bugs data
bugs.df<-read.csv("bugs.csv")
#Coerce it into a "BMI" data object
bugdata <- BMI(bugs.df)

#Subsample to 500 individuals and aggregate
bugdata.samp <- sample(bugdata)
bugdata.agg <- aggregate(bugdata.samp)

#Calculate metrics at SAFIT Level 1
metrics <- BMIall(bugdata.agg, effort=1)
```

Note: Users who have access to the SWAMP Reporting Module should use that tool instead.

Taxonomist over-rides of distinct taxa designations

Taxonomist over-rides of distinct taxa designations are no longer recommended for standard CSCI scoring. The CSCI calculator does not correctly score samples if the designations are at better resolution that SAFIT Level 1. That is, the calculator inlcudes taxa in richness estimates that should be aggregated to a higher taxonomic level (such as any genus, tribe, or subfamily Chironomidae that has been indicated as distinct). Because richness estimates appear in both the numerator and denominator of several metrics in the MMI, scores may be incorrectly inflated or deflated (although the latter is more common). We recommend leaving Distinct blank in all data inputs, without over-riding the automated distinct taxon designation process.

Need more help?

Join the CSCI users listserve by emailing Raphael Mazor (raphaelm@sccwrp.org). This listserve will eventually be replaced by one maintained by SWAMP.

Section 3: Cautions on Score Interpretation

Unusual Environmental Settings

Most wadeable streams can be accurately scored with the CSCI (including some nonperennial streams). However, the validity for sites from unusual environmental settings is unknown. Although indices based on predictive models typically flag sites with predictor data outside the experience of the model using a chi-square test, we do not endorse this approach, and have not included it in the `CSCI` package. Instead, we recommend a case-by-case approach to evaluating the applicability of the tool in unusual environmental settings. Data about reference sites provided by the `loadRefData()` function may help determine if a test site represents an unusual environmental setting.

Samples with Low Counts

Samples with low bug counts may have erroneously depressed CSCI scores. We have not established a minimum count of bugs for validating the CSCI, but as a rule of thumb, scores that are within 10% of the specified sample size (i.e., at least 450 individuals for the pMMI, and 360 unambiguous individuals for the O/E) may be used for most applications of the CSCI. Smaller counts may be appropriate for certain applications.

Samples with Many Ambiguous Individuals (e.g., all midges IDed to family)

Samples with many ambiguous individuals typically occur when early instars that cannot be reliably identified are abundant, or when samples were not originally taken to the desired level of taxonomic resolution (e.g., samples were identified to SAFIT Level 1). In the former case, both the O/E and pMMI may be depressed, even if the total number of individuals is very high. In the latter case, the O/E may be depressed, although the pMMI should be unaffected. Although no criteria have been established for evaluating the impacts of ambiguous organisms on the CSCI, we recommend evaluating both the Pcnt_Ambiguous_Individual and Pcnt_Ambiguous_Taxa values when interpreting scores.

Scoring of samples identified to a SAFIT Level 1 is not recommended in most circumstances. If samples are archived, the best solution is to get midges identified to subfamily by a taxonomist who participates in SAFIT. If this is not feasible, your next best option is to calculate the range of possible CSCI scores. The lowest possible score is estimated by calculating the CSCI with all midges left at Chironomidae. The highest possible score is estimated for each sample as follows:

1. Go to Suppl1_OE, and count up the number of midge subfamilies (i.e., Chironominae, Diamesinae, Orthocladiinae, Podonominae, Prodiamesinae, and Tanypodinae) that are expected in a given sample (i.e., CaptureProb ≥ 0.5) but that are also absent (i.e., MeanObserved = 0).
2. Go to the core report, and add the number from step 1 to O for that sample. This estimates a maximum value for O.
3. Estimate the maximum O/E by dividing the estimate from step 2 by E.
4. Estimate the maximum CSCI by adding the new maximum O/E estimate from step 3 to the MMI and dividing by 2.

In some cases, the range of possible CSCI scores may be small enough that decisions may be made with existing data (for example, if the highest possible score is below a target threshold, it may be determined that the site does not meet its objective). If the range is large enough to include an important threshold, it is strongly recommended that samples be sent to a midge taxonomist rather than using the estimation approach described here.

Samples Dominated by Oligochaetes

Samples dominated by Oligochaeta may end up failing to get scores for the pMMI.